GeoMeTree - Geodesic Metric on Trees

Manual Version 1.1 (August 21, 2009)

Anne Kupczok
Center for Integrative Bioinformatics Vienna,
Max F. Perutz Laboratories,
Dr. Bohr-Gasse 9/6, A-1030 Vienna, Austria.
email: `anne.kupczok @ mfpl.ac.at`

Arndt von Haeseler
Center for Integrative Bioinformatics Vienna,
Max F. Perutz Laboratories,
Dr. Bohr-Gasse 9/6, A-1030 Vienna, Austria.
email: `arndt.von.haeseler @ mfpl.ac.at`

Steffen Klaere
Center for Integrative Bioinformatics Vienna,
Max F. Perutz Laboratories,
Dr. Bohr-Gasse 9/6, A-1030 Vienna, Austria.
email: `steffen.klaere @ mfpl.ac.at`

Contents

1 Legal Stuff

2 Introduction

GeoMeTree is a computer program to compute the geodesic distance and the corresponding path between two weighted phylogenetic trees on the same leaf set as defined in Billera et al. (2001). The algorithm is described in Kupczok et al. (2008).

3 Installation

The command-line program is freely available from http://www.cibiv.at/software/geometree. It is written in python and should run on every computer with python version 2.4 or newer. Python can be downloaded from `http://www.python.org/`. Please note that different versions for python 2.x and python 3.x exist .

First unzip `GeoMeTree-1-1.zip` (for python 2.x) or `GeoMeTree3-1-1.zip` (for python 3.x). Then change into the directory `GeoMeTree` and type `python GeoMeTree.py [options]` to run the program.

4 Algorithm

The Algorithm is described in detail in Kupczok et al. (2008). Briefly, it computes the shortest path between two weighted phylogenetic trees in tree space. The tree space for a given number of leaves n is a subset of $\mathbb{R}_+^{2^{n-1}-1}$, where each unit vector corresponds to one of the $2^{n-1}-1$ possible splits. For every point in tree space, all components with a non-zero entry must refer to pairwise compatible splits. In general, the Euclidean path between two topologies traverses points which are not legal by this definition. On the other hand, the geodesic path traverses only legal points and is the shortest path between two trees through tree space.

We find this path by efficiently enumerating all possible topologies which could be traversed by the path. They are arranged in a directed acyclic graph (DAG), such that starting and ending topologies refer to the two given topologies.

To determine the exact progression of a path in tree space, the path is parameterized with constant speed to yield a function g (Vogtmann, 2003). $g(0)$ is the point in tree space corresponding to the first tree and $g(1)$ is the point in tree space corresponding to the second tree. A path is piecewise linear und changes its gradient only when it switches orthants at the so-called transition points. These transition points therefore fully determine a path between two weighted trees.

The algorithm of enumerating the topologies is exponential in the number of different splits between the two given topologies. However, the structure of the space allows us to provide certain improvements. When the DAG is build some topologies and paths are not enumerated if they cannot yield legal transitions. Further, the trees can be decomposed. In particular, the topologies are decomposed on their common edges and the paths through these decomposed topologies are computed independently. After calculating the individual paths these can be assembled to obtain the path for the full trees. Then all individual computations are only exponential in the number of the different splits between these smaller trees. This is a substantial reduction of computing time especially for biologically reasonable trees which are expected to share some splits.

5 Command-line options

Run `python GeoMeTree.py -h` to print out a short description of available options:

```
usage: GeoMeTree.py [options]

options:
  -h, --help            show this help message and exit

  Input options:
    -f INFILE, --file=INFILE
                        Name of input file, all pairs of trees in the file are
                        evaluated (no default!)

  Tree lengths options:
    -n, --norm          Normalize the branch length vectors to norm 1
                        (default: no normalization)
    -t, --term          Ignore branch lengths of terminal splits (then also
                        ignored in normalization, default: terminal branch
                        lengths considered)

  Algorithmic options:
    -a, --approx        Compute only the approximations, not geodesic path
    -d, --decomp        Do not decompose the trees (default: decomposition
                        when possible)

  Output options:
    -v HEADER, --header=HEADER
                        Header name of output file(s) (default: 'pair', then
                        files pair_i_j are generated for each pair)
```

5.1 Examples

```
python GeoMeTree.py --file=examples/example.trees --norm
python GeoMeTree.py -f examples/suborth.trees -v suborth
```

To test whether the program runs on your system you can test these commands and compare whether the resulting files `pair_1_2`, `pair_1_3`, `pair_2_3` and `suborth_1_2` equal those in the `example`-directory.

5.2 Input options

`-f INFILE, --file=INFILE`

`INFILE` is the name (and path) of the input file. All pairs of trees are evaluated and an output file will be generated for every pair. The input trees will always be interpreted as unrooted. Please insert a dummy root taxon in your tree if you want to evaluate rooted trees. E.g. the rooted tree `((A:1,B:1):1,(C:1,D:1):1)` would be interpreted as equivalent to the unrooted tree `((A:1,B:1):2,C:1,D:1)`. But you can add the root like this to your input trees: `((A:1,B:1):1,(C:1,D:1):1,R:0)`.

5.3 Tree lengths options

`-t, --term`

Every tree is associated with a branch lengths vector containing all weights in the tree. Only if this option is activated, the terminal branch lengths are discarded from the vector and thus from all further computations.

`-n, --norm`

Normalize the branch lengths vector of each tree to norm 1. This has the consequence that each tree has the same distance from the origin. The normalization lessens the impact of long branches on the geodesic distance thus and increases the impact of topological differences between trees.

5.4 Algorithmic options

`-a, --approx`

With this option, the geodesic distance is not computed. Instead, the programm returns its approximations, the branch-score distance and the cone distance. See Amenta et al. (2007) and Kupczok et al. (2008) for a definition of the approximations.

`-d, --decomp`

By default, the trees are decomposed on the common inner edges and the paths through the parts are computed independently. This does not affect the geodesic path (Vogtmann, 2003), but the number of enumerated paths is substantially decreased. This has a positive effect on the computing time and thus, using decomposition is highly recommended.

But there are several cases where decomposition has to be turned off: First, the upper bound changes: not the cone path but the decomposed cone path is computed (see section 6.2). If you are interested in the cone path which goes through the consensus of both trees, decomposition has to be turned off. Second, less paths are enumerated with decomposition. Thus, if you are interested in the complete number of possible paths between two trees, you should be aware, that the number of paths is different with and without decomposition. With decomposition only a lower bound of the number of paths can be computed. This bound is the product of the number of paths through the individual decompositions.

5.5 Output options

`-v HEADER, --header=HEADER`

An output file which starts with `HEADER` is generated for each pair. It lists the trees, the results for the single decomposition (if applicable) and the overall results.

6 Output specification

6.1 Without decomposition

The following example shows the output of the program if decomposition has been turned off (see `examples/suborth_1_2`):

```
2 Trees of 6 taxa given:
T1=((A:0.1,B:0.1):0.1,(C:0.1,D:0.1):0.2,(E:0.1,F:0.1):0.3);
T2=((A:0.1,C:0.1):0.9,(D:0.1,F:0.1):0.5,(B:0.1,E:0.1):0.6);

------------------------------------------------------------------------------

Splits only in T1:
1 A*B 0.100000
2 C*D 0.200000
3 E*F 0.300000

Splits only in T2:
4 A*C 0.900000
5 D*F 0.500000
6 B*E 0.600000

Splits common to both trees and branch length in T1 and T2:
```

```
7 F 0.100000 0.100000
8 E 0.100000 0.100000
9 D 0.100000 0.100000
10 C 0.100000 0.100000
11 B 0.100000 0.100000
12 A 0.100000 0.100000

The geodesic path:
t L R
0.000000 --
0.199008 1*2 4
0.277514 3 5*6
1.000000 --

Transition points:
Splits:     (      1,      2,      3,      4,      5,      6,     10,     11,     12,      7,      8,      9 )
g(0.0000) = ( 0.1000, 0.2000, 0.3000, 0.0000, 0.0000, 0.0000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000 )
g(0.1990) = ( 0.0000, 0.0000, 0.0849, 0.0000, 0.0000, 0.0000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000 )
g(0.2775) = ( 0.0000, 0.0000, 0.0000, 0.0882, 0.0000, 0.0000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000 )
g(1.0000) = ( 0.0000, 0.0000, 0.0000, 0.9000, 0.5000, 0.6000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000, 0.1000 )


Results for the different splits:
Branch score 1.249000
Geodesic distance 1.559201
Cone distance 1.565803

Results for all splits:
Branch score 1.249000
Geodesic distance 1.559201
Cone distance 1.565803

Complete graph enumerated: 2 path(s) found!

Computation time for geodesic path: 0.0000 s
```

The first section of the output lists the splits which are different between the trees and which are common in both trees. A split is a bipartition of the taxon set. In the output, it is given by the set of taxa on one site of the split, e.g. `A` is the terminal edge partitioning the taxon `A` from the rest and `A*B` is the inner edge partitioning `A` and `B` from `C`, `D`, `E` and `F`. A unique number is assigned to every split. Further, the branch lengths are given for every split. If normalization is activated, these lengths may differ from the lengths in the input tree.

A path between two weighted trees can be unambiguously described by the splits which are extended and shrunken at the same point t in time (for details see Kupczok et al., 2008). Between these transitions the path is linear. Thus the geodesic path is given by the transition times t and the numbers referring to the corresponding splits. L are the "left" splits from `T1` and R are the "right" splits from `T2`. If more than one split is joined at a time, `*` indicates the joining of these splits.

In addition, the path is given by the transition points. These are the points at which the topologies change, i.e. at a transition point, the corresponding splits in L and R have length zero.

Finally, the geodesic distance and its approximations are given for two cases: first, when the branch lengths of the common splits are ignored ("Results for the different splits") and second for all splits.

6.2 With decomposition

The following example shows the output of a computation with decomposition (see `examples/pair_1_2`):

```
2 Trees of 6 taxa given:
T1=(((A:0.1,B:0.2):0.6,C:0.1):0.7,D:0.3,(E:0.3,F:0.4):0.5);
T2=(((A:0.1,C:0.2):0.5,B:0.1):0.6,E:0.3,(D:0.3,F:0.4):0.7);

Tree vectors have been normalized to norm 1 !!!

Trees have been decomposed at common splits, the following dummy taxa are used:
d1 A*B*C
d2 E*D*F

--------------------------------------------------------------------------------

Results for Decomposition No. 1:

Splits only in T1:
1/1 D*d1 0.408248

Splits only in T2:
1/2 D*F 0.571548

Splits common to both trees and branch length in T1 and T2:
1/3 d1 0.571548 0.489898
1/4 F 0.326599 0.326599
1/5 E 0.244949 0.244949
1/6 D 0.244949 0.244949

The geodesic path:
t L R
0.000000 --
0.416667 1/1 1/2
1.000000 --

Transition points:
Splits:     (    1/1,    1/2,    1/3,    1/4,    1/5,    1/6 )
g(0.0000) = ( 0.4082, 0.0000, 0.5715, 0.3266, 0.2449, 0.2449 )
g(0.4167) = ( 0.0000, 0.0000, 0.5375, 0.3266, 0.2449, 0.2449 )
g(1.0000) = ( 0.0000, 0.5715, 0.4899, 0.3266, 0.2449, 0.2449 )


Results for the different splits:
Branch score 0.702377
Geodesic distance 0.979796
Cone distance 0.979796

Results for all splits:
Branch score 0.707107
Geodesic distance 0.983192
Cone distance 0.983192

Complete graph enumerated: 1 path(s) found!
```

```
Computation time for geodesic path: 0.0000 s

--------------------------------------------------------------------------------

Results for Decomposition No. 2:

Splits only in T1:
2/1 A*B 0.489898

Splits only in T2:
2/2 A*C 0.408248

Splits common to both trees and branch length in T1 and T2:
2/3 C 0.081650 0.163299
2/4 B 0.163299 0.081650
2/5 A 0.081650 0.081650

The geodesic path:
t L R
0.000000 --
0.545455 2/1 2/2
1.000000 --

Transition points:
Splits:     (    2/1,    2/2,    2/3,    2/4,    2/5 )
g(0.0000) = ( 0.4899, 0.0000, 0.0816, 0.1633, 0.0816 )
g(0.5455) = ( 0.0000, 0.0000, 0.1262, 0.1188, 0.0816 )
g(1.0000) = ( 0.0000, 0.4082, 0.1633, 0.0816, 0.0816 )


Results for the different splits:
Branch score 0.637704
Geodesic distance 0.898146
Cone distance 0.898146

Results for all splits:
Branch score 0.648074
Geodesic distance 0.905539
Cone distance 0.905539

Complete graph enumerated: 1 path(s) found!

Computation time for geodesic path: 0.0000 s

--------------------------------------------------------------------------------

The complete geodesic path:
t L R
0.000000 --
0.416667 1/1 1/2
0.545455 2/1 2/2
1.000000 --

Transition points:
```

```
Splits:     (    1/1,    2/1,    1/2,    2/2,    1/3,    1/4,    1/5,    1/6,    2/3,    2/4,    2/5 )
g(0.0000) = ( 0.4082, 0.4899, 0.0000, 0.0000, 0.5715, 0.3266, 0.2449, 0.2449, 0.0816, 0.1633, 0.0816 )
g(0.4167) = ( 0.0000, 0.1157, 0.0000, 0.0000, 0.5375, 0.3266, 0.2449, 0.2449, 0.1157, 0.1293, 0.0816 )
g(0.5455) = ( 0.0000, 0.0000, 0.1262, 0.0000, 0.5270, 0.3266, 0.2449, 0.2449, 0.1262, 0.1188, 0.0816 )
g(1.0000) = ( 0.0000, 0.0000, 0.5715, 0.4082, 0.4899, 0.3266, 0.2449, 0.2449, 0.1633, 0.0816, 0.0816 )

Results for the different splits:
Branch score 0.948683
Geodesic distance 1.329160
Decomposed cone distance 1.329160

Results for all splits:
Branch score 0.959166
Geodesic distance 1.336663
Decomposed cone distance 1.336663

Complete graphs enumerated: with independent decompositions 1 path(s) found!

Computation time for all geodesic paths: 0.0000 s
```

Dummy taxa are introduced into the decomposed trees because they are generated from the original trees by replacing a subtree by a dummy terminal node. Thus the split `D*d1` refers to the split `A*B*C*D` in the original tree.

Now, the splits are not named by numbers but have the form `i/j` where `i` is the number of the decomposition and `j` is the number of the split in the respective decomposition.

The results are first given for every decomposition separately and then for the complete trees. The format is the same as described in section 6.1.

But for the complete trees, there are two differences in the output compared to a computation without decomposition: First, not the number of paths, but the number of paths with independent decompositions is given. This is the product of the number of paths of the individual decompositions. It is a lower bound of the number of paths without decomposition. Second, not the 'cone distance' but the 'decomposed cone distance' is given. There, the local cone is computed for every decomposition separately. In the above example, the geodesic path equals the decomposed cone path, but the cone path is given by $L = \texttt{1/1} * \texttt{2/1}$ and $R = \texttt{1/2} * \texttt{2/2}$. Thus, the cone path always has only one transition point, whereas the decomposed cone path has one transition point for every decomposition. The cone path has been described to be a very good approximation of the geodesic distance (Kupczok et al., 2008), but the approximation is even improved by the decomposed cone path.

7 Version History

Version 1.1 Exception introduced if the trees contain too many different splits, then it is suggested to run the approximations. Furthermore a python3 version is now available.

References

Nina Amenta, Matthew Godwin, Nicolay Postarnakevich, and Katherine St. John. Approximating geodesic tree distance. *Inf. Process. Lett.*, 103(2):61–65, 2007.

Louis J. Billera, Susan P. Holmes, and Karen Vogtmann. Geometry of the space of phylogenetic trees. *Adv. Appl. Math.*, 27:733–767, 2001.

Anne Kupczok, Arndt Von Haeseler, and Steffen Klaere. An exact algorithm for the geodesic distance between phylogenetic trees. *J Comput Biol*, 15(6):577–591, 2008. doi: 10.1089/cmb.2008.0068.

Karen Vogtmann. Geodesics in the space of trees. Technical report, Cornell University, 2003.